PER
C=64
E
C=128

Numero 42
Lire 12.000

CONTIENE DISK CON DUE FACCIATE DI PROGRAMMI

Distributore: Marco A. & G. S.p.A.

3D GAMES CONSTRUCTION KIT

PER CREARE UN VIDEOGIOCO AD ALTO LIVELLO CON GRAFICA TRIDIMENSIONALE

GRUPPO LOGICA 2000

IN EDICOLA

COMPUTING VIDEOTECA

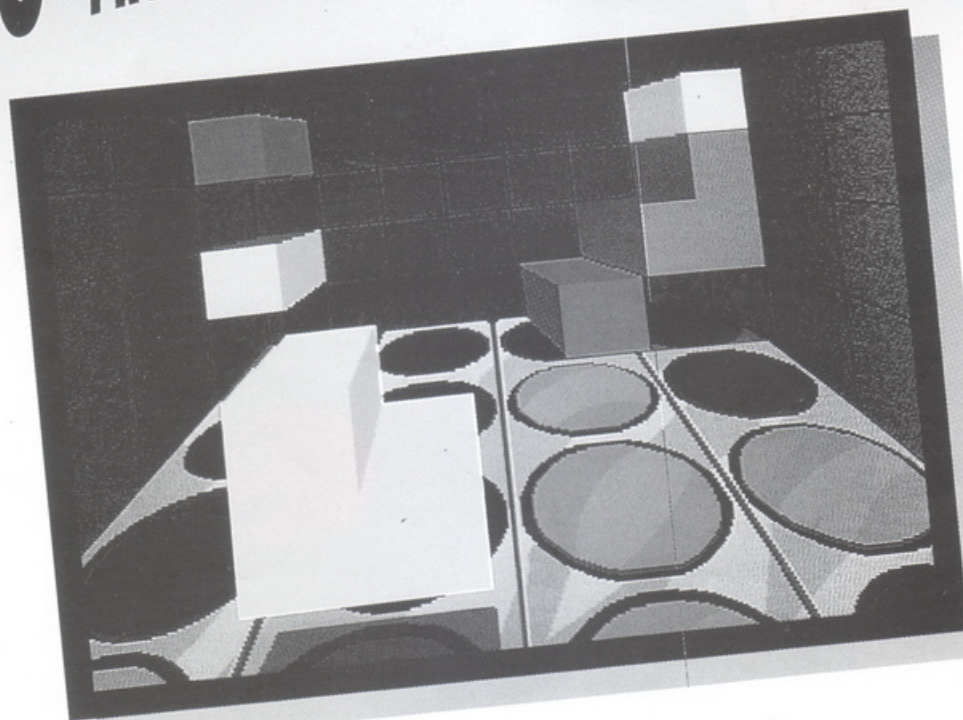
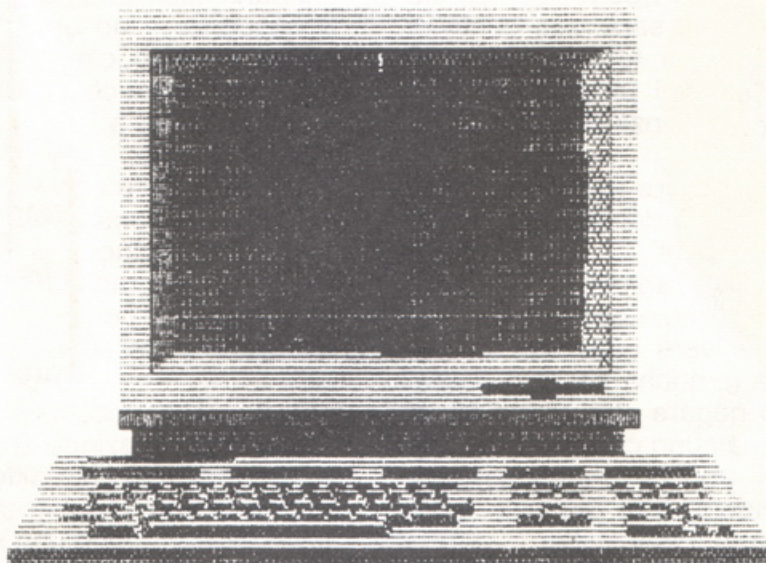
Nuova serie
Numero 42
Lire 8.000

L2 GRUPPO LOGICA 2000

PER C= 64 E C= 128

ABILITY GAMES

10 PROGRAMMI PER METTERE ALLA PROVA LA VOSTRA ABILITÀ

- SHANGHAI
- ARRANGEMENT TILES
- MOLECULE
- SOLITARY
- QUOIT
- COLOUR BALL
- 64 TILES
- COLOUR TILES 2
- CONFUSEDLY
- FORZA 4

PER UN VIDEOGIOCO AD ALTO LIVELLO CON GRAFICA TRIDIMENSIONALE

INTRODUZIONE

Se siete degli appassionati di videogames, probabilmente vi sarete spesso chiesti quanto bravi siano gli ideatori di quei fantastici giochi che sono in vendita per il Commodore 64.
In effetti queste persone, oltre ad avere una fantasia fuori dal comune, sono dei veri e propri maghi del computer, conoscono il Commodore 64 in ogni minimo particolare e per loro Basic e linguaggio macchina non hanno misteri.

Se poi tenete presente che per i giochi più complessi la produzione non è affidata ad un solo programmatore, ma ad un team di informatici, ognuno dei quali (se non si tratta addirittura di un gruppo) si occupa di un particolare aspetto del programma.
C'è chi tratta solo la parte sonora del videogioco, chi si occupa della grafica, chi della strategia eccetera, così da spezzare i compiti in modo che i tempi di produzione vengano ridotti al minimo.
Se così non fosse, se un programma fosse af-

FLOPPY 64 n.42/1991
Periodico mensile di cultura informatica e programmi. Edizioni Gruppo LOGICA 2000 srl - Direzione, amministrazione e redazione: Via Francesco Sforza, 14 Milano - Registro stampa al Tribunale di Milano n. 145 del 5/3/1988 - Distribuzione A.& G. Marco Via Fortezza, 27 Milano - Stampa Aligraf Milano. Hanno collaborato: Stefano Currò, Giusy Quattrini, Alfredo Onofrio. Direttore responsabile: Antonio Lucarella. Ufficio Tecnico: telefono (02) 3314868.

fidato ad una sola persona, per quanto brava e geniale possa essere, per quanto veloce e corretta nello scrivere i programmi e per quanto possa conoscere l'assembler o il linguaggio macchina, non potrebbe finire un lavoro in tempo utile perchè entri nel mercato nelle scadenze prefissate.
Del resto se qualcuno di voi ha mai provato a realizzare un videogioco in Basic che non sia ad un livello elementare, si sarà reso conto di quanti particolari si debba tenere conto, di quante siano le variabili del programma e di come sia difficile gestirle senza perderci la testa.
E poi i risultati non sarebbero nemmeno paragonabili con quelli ottenuti dai realizzatori dei più famosi 'supervideogiochi'.
Questo succede per varie ragioni, la prima sta nel fatto che quelle persone, come già più volte ripetuto, sono delle vere superstar del videogame, la seconda (e anche più importante, perchè nessuno può negare che anche tra di voi ci possa essere qualcuno con le medesime qualità, magari non ancora compiutamente espresse) è che i videogiochi più belli sono tutti realizzati in linguaggio macchina.
Ma che cos'è il linguaggio macchina si chiederanno quelli che solo da poco si sono affacciati al mondo del computer.
È presto detto, se avete la pazienza di seguire il mio discorso (per altro molto semplificato).

Il linguaggio macchina, questo sconosciuto

Quando voi digitate un comando Basic, come ad esempio

IF A > 0 THEN GOSUB 350

in pratica parlate al computer in una lingua chiamata Basic, che assomiglia vagamente all'inglese (le parole IF, THEN e GOSUB sono in effetti inglesi), ma che non può dirsi propriamente inglese.
In realtà voi dite al computer una cosa di questo genere:
'Se il valore contenuto nella variabile A è maggiore di 0 (ma non uguale!!), allora passa ad eseguire un altra parte di programma (un sottoprogramma) che inizia alla riga 350'.
Tutto questo è stato detto in linguaggio naturale e più precisamente in italiano.
Certo per noi è molto più semplice istruire il computer con una lingua che conosciamo bene come l'italiano, che dover ricorrere a certe frasi strane pseudo-matematiche e per di più in una lingua straniera, ma vedete, l'italiano, come un qualsiasi altro *linguaggio naturale* è molto complesso, comprende molte parole e il computer le dovrebbe memorizzare tutte per comprendere le nostre istruzioni: tutto questo richiederebbe un grosso spreco di memoria.
E poi un linguaggio naturale si presta molto ad incomprensioni ed imprecisioni che solo noi ...umani riusciamo a superare grazie a delle conoscenze e ad una perspicacia che difficilmente è introducibile o insegnabile ad un calcolatore; senza contare che l'ambiguità del linguaggio porta a volte anche noi stessi, a dei fraintendimenti.
Il più delle volte capiamo dal contesto ciò che il nostro interlocutore ci vuol dire, ma un computer, che traduce alla lettera ciò che gli si dice, non è in grado di estrapolare il significato della frase dal contesto.
Facciamo un esempio, in italiano esiste la frase:

'Una vecchia porta la sbarra'

Questa dichiarazione si presta a due interpretazioni notevolmente differenti:
a) Una donna vecchia sta camminando, mentre tiene in mano una sbarra
b) Una porta vecchia sta sbarrando la strada ad un persona o cosa di genere femminile (ad esempio una ragazza)

L'ambiguità nasce dal fatto che alla paroletta 'la' possono esser dati due significati, o di pronome personale femminile, o di articolo determinativo femminile.
Il fatto è che noi dal resto della frase riusciamo sempre (o quasi) a capirne il senso, ma per il calcolatore questa operazione è tremendamente complicata, se non impossibile.
Casi di questo tipo non sono poi così isolati e ne esistono in tutte le lingue ad esempio ancora per l'italiano la frase:

'I polli sono pronti per il pranzo'

o per chi sa l'inglese:

'He saw that gasoline can explode'

Traduzione 1) 'Vide che la benzina può esplodere'
Traduzione 2) 'Vide quel bidone di benzina esplodere'

In effetti un lavoro così apparentemente semplice come il comprendere le frasi di un'altra persona, necessitano di una notevole dose d'intelligenza per essere eseguite, anche se noi non ci facciamo caso, infatti il computer, velocissimo a fare i calcoli, ma poco perspicace, è poco intelligente (se vogliono azzardare ad associare le due cose), e questo discorso

contrasta notevolmente con quella che le convinzioni popolari, che tendono a mitizzare il computer e a ritenerlo una macchina pensante a tutti gli effetti ed in grado di eseguire qualsiasi 'lavoro' mentale.
Studiosi e ricercatori stanno facendo grossi sforzi, ma sembra che la comprensione del linguaggio naturale, da parte del computer, sia un problema veramente difficile da risolvere, tanto da rientrare in quella disciplina dell'informatica chiamata *intelligenza artificiale*.
Dopo aver parlato così a lungo del linguaggio naturale (e forse anche troppo, ma l'argomento è interessante, oltre che curioso), abbiamo appurato che non possiamo, almeno per il momento, comunicare con il computer nella maniera a noi più congeniale; proviamo allora a vedere se ci conviene colloquiare col computer nella maniera a lui più consona.
Il computer non capisce le parole, nè i suoni, in realtà noi comprendiamo questi messaggi perchè abbiamo un senso (l'udito) predisposto al rilevamento di questi fenomeni.
Il calcolatore non possiede il senso dell'udito, ma ciò che possiede è la capacita di rilevare la presenza o l'assenza di energia elettrica all'interno dei suoi circuiti.
I sensi del calcolatore sono limitati a questa comprensione e se vogliamo comunicare nella 'sua lingua' dobbiamo adeguarci a ragionare in una maniera diversa.
Volendo tradurre in simboli il linguaggio del computer, si può tradurre il passaggio di corrente in un circuito con un 1 e l'assenza del segnale con uno 0; ecco allora che le parole comprese da un computer sono formate da sequenze di uni e di zeri, dette anche cifre binarie.
Ad ogni sequenza diversa di cifre binarie corrisponde una parola dal significato diverso per il calcolatore; ecco allora che se noi vogliamo comunicare delle istruzioni ad un calcolatore nella sua lingua, dobbiamo costringerci ad imparare il significato delle parole binarie.
Questo è nella realtà un lavoro improbo e notevolmente difficile; per prima ragione perchè è più facile sbagliarsi e secondo perchè con queste parole, fatte di uni e di zeri, si deve seguire una certa grammatica, si devono cioè seguire certe regole con cui costruire delle istruzioni binarie per il computer.
Ebbene questo è il *linguaggio macchina*.
Voi direte ma perchè per realizzare un bel gioco si deve per forza 'parlare' al Commodore in linguaggio macchina quando c'è il Basic che, viste le prospettive, non ci sembra poi così male.
Facciamo un esempio illuminante.
Supponiamo che voi andiate in Israele, là si parla ebraico, perciò, in teoria, per comunicare con un israeliano dovremmo imparare questa nuova lingua, peraltro piuttosto difficile dato che l'alfabeto è totalmente diverso dal nostro.
Pensate allo sconforto che provereste solo a trovarvi attorno scritte incredibili ed incomprensibili.
D'altra parte gli israeliani, in maggioranza, non conoscono l'italiano e il problema d'intendersi diventa così drammatico.
L'unica soluzione che resta è quella di servirsi di una terza lingua, conosciuta da entrambi, ma che entrambi, in diversa misura, facciamo fatica a parlare: l'inglese.
Un colloquio a questo punto può essere instaurato, procederà a tentoni, a seconda che le conoscenze d'inglese siano più o meno approfondite nei due interlocutori (basta che uno di essi fatichi a parlare l'inglese, perchè il discorso diventi irto di difficoltà), ma senza dubbio si è trovata una buona soluzione.
Ebbene lo stesso avviene tra un uomo e un calcolatore: la macchina non può parlare il linguaggio naturale, l'uomo non vuole fare quella tremenda fatica che è imparare il linguaggio macchina, e così si ricorre ad una lingua intermedia che tutti e due conosciamo: il Basic (o qualsiasi altro linguaggio di programmazione, come il Cobol, il Pascal o il C).
Il Basic non è la nostra lingua e proprio per questo spesso sbagliamo ad impartire gli ordini al computer, questi invece non sbaglia mai (di questo si può essere certi) perchè, come per gli israeliani l'inglese è la seconda lingua, il computer viene istruito, con un programma scritto in linguaggio macchina, a tradurre il Basic in istruzioni binarie.
Questo programma viene chiamato *interprete*.
Questo lavoro di traduzione è però abbastanza dispendioso per il calcolatore, perciò le istruzioni che noi diamo al computer non vengono eseguite con la stessa prontezza che se avessimo usato istruzioni in linguaggio macchina; è un pò come se noi volessimo che l'israeliano ci compisse un servizio il più velocemente possibile, ma, malgrado il nostro inglese sia corretto (e già, perchè se il programma in Basic non è corretto il computer si ferma subito), l'uomo non si muove sino a che non ha tradotto nella sua lingua i nostri comandi.
In definitiva i programmi in linguaggio macchi-

na sono molto più veloci di quelli scritti in Basic, ma anche più difficili da scrivere.
In un campo come quello dei videogames dove la velocità d'azione vuol dir molto (se non tutto), è indispensabile conoscere il linguaggio macchina come le proprie tasche, escludendo i poveri mortali dall'elite dei grandi disegnatori di videogiochi.
Un'ultima cosa a proposito del Basic.
Quelli che tra voi sono stati più attenti si saranno chiesti: 'Ma perchè se ci possiamo scegliere una lingua intermedia tra il linguaggio macchina e l'italiano scegliamo proprio il Basic che è così strano e così simile all'inglese, non ci potrebbe essere un linguaggio più... italianeggiante ?'
La risposta è semplice, noi usiamo il Basic perchè lo hanno inventato gli americani che sono molto più all'avanguardia di noi in fatto d'informatica, volendo potremmo tradurre tutte le parole inglesi del Basic in italiano ed avere istruzioni del tipo:

SE A > 0 ALLORA SOTTOPROG 350

ma chi dovrebbe compiere questo lavoro è un esperto informatico, e per un esperto informatico l'inglese è il pane quotidiano, deve continuamente consultare manuali in inglese, ha studiato per anni su libri scritti in inglese e perciò non si spaventa a dover programmare in un Basic, che è così simile alla lingua inglese.
Perciò se da adulti pensate di voler lavorare nel campo dell'informatica non trascurate lo studio dell'inglese, perchè una buona conoscenza di quella lingua sarà fondamentale.
E poi si sa che gli italiani sono americanofili e, quando possono, amano mettere insieme qualche parola d'inglese; quindi quando devono programmare non si lamentano troppo se il Basic assomiglia molto all'inglese e non si scomodano a farne una versione italiana, come hanno invece fatto i francesi.

Il programma nel dischetto

Il programma che vi presentiamo in questo numero vi permetterà di diventare voi stessi dei veri e propri 'mostri' nella creazione di videogiochi, senza però costringervi ad apprendere il linguaggio macchina o qualsiasi altro linguaggio di programmazione.
Il programma consiste in un kit di utilities per la generazione di videogames, e, sebbene non sia questo il primo programma del genere, offre delle possibilità clamorose a chiunque abbia abbastanza inventiva da ideare un videogioco.
Un potente sprite editor vi permetterà di generare gli sprite per il vostro programma, concedendovi inoltre una vasta gamma di strumenti per la manipolazione e la colorazione degli oggetti da voi creati.
Un generatore di suoni vi consentirà di definire gli effetti speciali del vostro gioco, come ad esempio gli scoppi, gli spari o gli scontri tra sprites.
Altri strumenti di lavoro vi serviranno a definire gli sfondi i nemici o anche i semplici oggetti che volete inserire nel vostro videogames personale.
Ma la novità più eclatante di questo generatore di giochi è quello di poter creare un ambiente di gioco tridimensionale!
Questa è una novità assoluta nel campo dei generatori di giochi, infatti sino ad ora eravamo stati abituati a vedere sì degli ottimi e potenti programmi del genere, ma tutti quanti consentivano la creazione di giochi in ambiente bidimensionale e, in questo senso, il programma che vi presentiamo è una vera e propria primizia.
Se poi avete avuto la cattiva esperienza di imbattervi in un precedente generatore di giochi che non vi ha soddisfatto perchè troppo complicato da usare e troppo poco flessibile, non dovete partire prevenuti, perchè vedrete anche voi che questo nuovo programma è molto semplice da usare e di possibilità ne offre a migliaia, basta avere un pò pazienza e di tempo, perchè costruire un videogame ad alto livello non è mai un gioco da ragazzi che può essere fatto in quattro e quattr'otto.
Vi starete chiedendo se tutta questa facilità nello creare dei giochi viene alla fine pagata con una certa lentezza del codice generato.
Nient'affatto i giochi creati con questo programma sono velocissimi ed offrono una così alta gamma di effetti da far concorrenza ai più bei videogame della vostra collezione.
In effetti i videogames prodotti con questo generatore sono molto meglio di molti dei giochi che, magari, avete comprato a caro prezzo, ma, soprattutto, ciò che stupisce è la rapidità con la quale è possibile generare videogames coloratissimi e dai mille risvolti.
Ma prima di proseguire con la spiegazione precisa di come si usa il programma, cerchiamo di identificare il gioco tipo che può essere creato con questo generatore.
Forse non ci avete mai pensato, ma esistono

diversi generi di videogames: quelli sportivi, quelli di simulazione, quelli di abilità ecc.
Un genere molto sfruttato (e al quale possono esser fatti risalire anche molti altri generi) e quello dello ... 'spara e fuggi'.
In questa categoria cadono ad esempio quasi tutti i giochi spaziali, in cui c'è un'astronave che viene attaccata da orde di navicelle nemiche e il giocatore deve sapersi destreggiare tra missili, meteoriti e scontri accidentali contro le navi nemiche.
Questi giochi sono a schermo fisso e sono i nemici ad entrare nel campo di gioco per uscirne solo quando vengono colpite; esistono però varianti in cui la nostra astronave vaga nello spazio ed avanza continuamente alla ricerca di una meta il più delle volte non ben definita.
Altri esempi dello spara e fuggi sono quei giochi in cui il nostro eroe deve passare in mezzo ad un campo nemico, tra gli spari degli avversari e trabocchetti di ogni genere, come accade, per esempio, in Commando o in Outlaw.
Il denominatore comune in tutti questi giochi sta nel fatto che il giocatore manovri un oggetto con la facoltà di sparare a più non posso, e così in effetti deve fare se vuole salvare la pellaccia contro un numero impari di nemici.
Generalizzando si può dire che, in giochi di questo genere, esiste sempre un omino o un'astronave che viene controllata col joystick; questa astronave ha un raggio di azione più o meno limitato: può svariare lungo tutto lo schermo o può anche rimanere fissa sull'ultima linea in basso del monitor, in genere ha però un numero illimitato di colpi da sparare contro dei nemici.
A loro volta i nemici possono essere di vari tipi, ma, inevitabilmente, devono essere presenti in questo tipo di giochi; a volta i nemici debbono essere colpiti più volte per essere eliminati, altri, se colpiti, non vengono mai eliminati, ma la loro azione viene rallentata, altri ancora regalano speciali Bonus oppure, in caso di scontro, ci fanno perdere un'astronave, ma non scompaiono di scena.
Altra caratteristica di questi giochi sono i livelli di gioco, via via sempre più difficili man mano che si prosegue; a volte vengono inframezzati dai cosiddetti *challenge stages* in cui il giocatore ha la possibilità di fare dei gran punti, senza però rischiare di perdere una vita.
Esiste infine uno sfondo che può essere lo spazio sconfinato o una base spaziale oppure il terreno sul quale il nostro uomo si muove in avanzamento tra mille difficoltà.
Lo sfondo è spesso molto colorato e presenta degli oggetti che ostacolano il passaggio dell'oggetto che comandiamo col joystick.
A volte il solo colpire uno di questi oggetti ci fa perdere una vita, altre volte ci sbarra semplicemente la strada ed altre volte ancora ci regala dei punti se raggiunti e 'catturati' dal nostro omino.
Ogni sparo, ogni scontro, ogni esplosione è caratterizzata da un effetto onoro associato, il più delle volte molto piacevole e fantasioso.
Infine il nostro uomo (o astronave) sembra muoversi in avanti perchè il video scrolla verticalmente; con questo generatore non è obbligatorio far scrollare lo sfondo, ma possono anche essere organizzati livelli di gioco dallo schema fisso, ma è certo che lo scroll da quel tocco di classe in più che mai guasta in un videogame.
Avrete oramai capito che questo programma 'generatore' di giochi offre innumerevoli possibilità e che certi trucchetti possono essere individuati solo dopo un pò di tempo che si è imparato ad usare il programma, perciò è meglio passare alla spiegazione di come va usato per realizzare il vostro primo videogioco.

3D GAMES

Per caricare il programma inserite il dischetto nel drive e digitate LOAD"LOADER",8,1 e lanciatelo con RUN. Seguite poi le indicazioni che compariranno sul video

Dopo una lunga introduzione passiamo alla spiegazione del programma.
Chiamare 'programma' questo prodotto è certamente riduttivo, in effetti si tratta più propriamente di un vero e proprio ambiente di sviluppo per la realizzazione di videogames.
Come è stato più volte rimarcato, la produzione di un videogioco non è attività semplice e richiede una conoscenza profonda del calcolatore per cui si realizza il gioco (nel nostro caso il Commodore 64).
Perciò il principiante non può sperare in poco tempo di diventare un perfetto programmatore, esperto al punto di poter costruire un videogame in grado di fare la concorrenza ai più noti giochi in vendita nei negozi specializzati.
Tuttalpiù si sarà capaci di programmare un poco in Basic e di realizzare un giochino semplice semplice capace di divertire per qualche oretta.
È chiaro che il solo fatto di essere riusciti a produrre un videogame, per quanto semplice e banale, ripaga enormemente gli sforzi compiuti e ci porta a non giudicare con eccessiva severità le pecche del nostro prodotto.
Dopo anni di studi ci si accosta timidamente al linguaggio Assembler o, addirittura, al linguaggio macchina ed allora le cose cambiano sensibilmente: questi sono gli strumenti adatti per realizzare un buon videogioco.
Ancora una volta però non si può entrare in competizione con i videogames venduti nei negozi; questi giochi infatti sono prodotti da una vera e propria equipe di programmatori che, con un oculato lavoro di squadra, si dividono il compito e realizzano prodotti strepitosi.
Questo discorso è necessario per farvi comprendere l'oscuro ed immane lavoro nascosto che questo programma svolge per voi.
In effetti con uno sforzo relativamente piccolo, anche voi potrete avere la soddisfazione di creare dei piccoli gioielli software, che niente hanno da invidiare ai giochi che potete acquistare nei negozi o cui potete giocare nelle migliori sale giochi.
In effetti ciò che il programma si limita a fare è quello di tradurre in istruzioni in linguaggio macchina i comandi e i controlli che voi indicherete durante la progettazione del gioco.
Questi comandi però verranno impartiti per lo più con dei semplici spostamenti del joystick e non dovendo imparare un difficile linguaggio di programmazione.
In effetti l'interfaccia tra voi e la memoria del Commodore 64 sarà il vostro fido joystick, con esso potrete indicare al computer quali debbano essere le condizioni di gioco, l'ambiente in cui la vostra astronave si muove, le situazioni che il giocatore dovrà affrontare.
È chiaro che un lavoro complesso non può essere ridotto al semplice maneggiamento di una leva del joystick, magari corredato da qualche pressione di tasto (altrimenti sarebbe una bugia che i videogames belli sono complessi da realizzare), anche con questo ambiente di sviluppo un certo sforzo viene richiesto.
Ma andiamo per gradi.
È logico che nella progettazione di un qualsiasi sistema si deve aver ben chiaro in partenza ciò che si vuole realizzare.
Non è possibile sedersi al tavolo, con il programma caricato nella memoria del Commodore 64, ed improvvisare la programmazione di un videogames (bello o brutto che sia).
Prima di far ciò vi consiglio di scrivervi in un foglio il progetto del gioco che state per realizzare.
Innanzitutto incominciate a pensare alla situazione in cui volete calarvi; con ciò intendo dire che dovete ideare un ambiente in cui inserire la vostra astronave, ad esempio un pianeta sconosciuto.
Provate ad immaginare come debba essere fatto: se ha montagne o pianure, valli strette o larghe eccetera eccetera.
Non è banale quello che vi sto dicendo, perchè se non compiete questo sforzo all'inizio, vi troverete poi a dover improvvisare e i risultati saranno catastrofici.
Nel pensare all'ambiente dovrete tenere conto che il gioco vi consente di creare scenari tridi-

mensionali!
Questa infatti è una delle caratteristiche più innovative del gioco; altri programmi di questo tipo sono stati creati in passato, ma mai nessuno dava la possibilità di creare paesaggi in 3D.
Una volta pensato l'ambiente in cui si debba muovere l'astronave, dovrete ideare le limitazioni di movimento dell'astronave nell'ambiente.
Ad esempio se la nave va a sbattere contro una roccia è inevitabile uno scoppio; ciò sembra naturale, ma in questo caso dovrete essere voi a deciderlo e a far sì che il programma si comporti secondo le vostre intenzioni.
Per assurdo potreste anche decidere che, in caso di urto, la vostra astronave acquista punti o guadagna una vita; è questo il bello della produzione di un videogame: si è i padroni assoluti della situazione e si decide da soli ciò che è plausibile o meno nella realtà del gioco.
Una volta decise le regole si deve pensare alla storia che sta dietro il gioco.
Infatti ogni videogame è governato da una storia, più o meno esplicita, che guida e spinge il giocatore a compiere certe azioni piuttosto che altre.
Per fare un esempio, in molti giochi spaziali un'astronave viaggia lungo un percorso pieno di navi nemiche, combatte contro queste navi e, se vince, prosegue il viaggio sino ad un misterioso pianeta (di solito teatro di un bonus stage).
Da qui la nostra astronave riparte per un nuovo, più lungo e pericoloso viaggio, e così via sino al termine della partita.
Senza questa fase di ideazione non può esistere un videogame, infatti la storia che sta dietro al gioco è ciò che più spesso invoglia a cimentarsi.
Inoltre una storia ben ideata contribuisce al successo e alla comprensione del videogame da parte del giocatore, che così sa sempre cosa deve fare e come agire.
È in questa fase che la vostra fantasia si deve sbizzarrire: trame piatte e senza verve possono essere genitrici di giochi scadenti.
La cosa più importante e che voglio ribadire, è che queste frasi ora descritte vengano realizzate a tavolino prima di mettervi al lavoro col computer: la realizzazione di un buon prodotto, da che mondo è mondo, è sempre stata preceduta da un'accurata fase di ideazione.
Così come queste fasi iniziali sono state distinte, anche il menu principale del programma ci riserva una distinzione per poterne eseguire la realizzazione tecnica.
In effetti dopo un breve caricamento del programma in memoria centrale, apparirà sullo schermo un menu:

1) EDITOR 1
2) EDITOR 2
3) COMPILATORE

PREMERE UN NUMERO PER SCEGLIERE

Le prime due parti del programma rispecchiano la realizzazione tecnica di una delle fasi di progettazione descritte sopra.
Ma andiamo per ordine e vediamole una alla volta.
Premetto che una trattazione dettagliata di ogni singola opzione richiederebbe uno spazio che va ben oltre le pagine di questa rivista, perciò cercherò di essere breve e schematico.

1) EDITOR 1

Questa parte di programma vi consente di realizzare lo scenario per la vostra storia.
L'ambientazione è in forma tridimensionale, perciò il programma vi mette a disposizione degli strumenti che vi consentono di modellare il paesaggio come più vi piace e di dare allo scenario un'immagine 3D.
Lo schermo è diviso in quattro parti:
1. La barra dei menu a tendina nella riga superiore dello schermo
2. Il tavolo di lavoro in cui potrete osservare come lo scenario del vostro videogame si modifica a seguito dei vostri colpi di... joystick.
3. Il pannello dei comandi che vi presenta gli strumenti necessari alla definizione dello scenario.
4. Il pannello di controllo che ripropone in pratica i controlli che altrimenti sarebbero accessibili attraverso il menu a tendina.

1. La barra dei menu
La barra sulla riga superiore dello schermo consta di due menu a tendina: FILE e AREA.
Alla barra dei menu si accede spostando verso l'alto la leva del joystick; a questo punto vedrete evidenziarsi uno dei titoli dei menu a tendina: premendo il fire il menu calerà sullo schermo evidenziandone le voci; spostando invece lateralmente la leva del joystick si passerà ad evidenziare il titolo del menù a tendina

seguente.
Con FILE vi si consente di operare sul disco, vale a dire di salvare o caricare il lavoro già fatto.
Con AREA, invece, ci si può spostare da un area all'altra.
Un'AREA è la porzione di paesaggio che è possibile mostrare sullo schermo; è chiaro però che uno scenario completo è composto da più aree accostate.
Con AREA si può saltare velocemente da un'AREA all'altra.
Sulla parte destra della barra menù è indicata la memoria ancora libera: vedrete che man mano che lavorate ed inserite dati tale valore diminuirà.

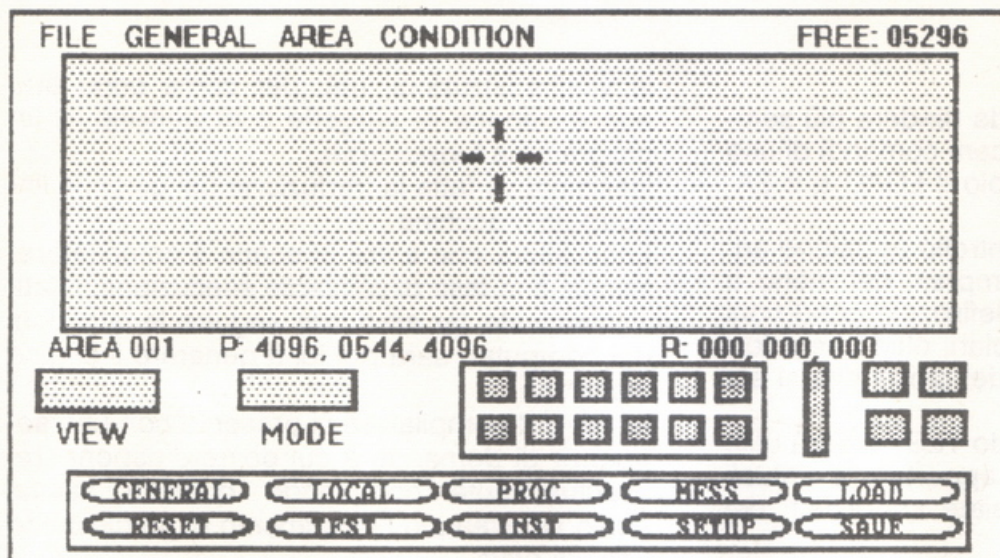
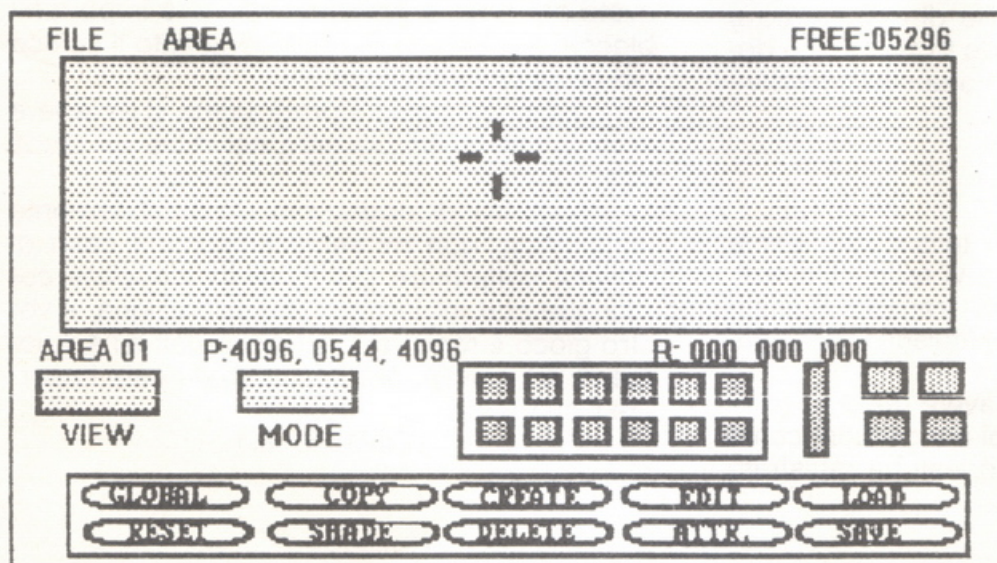
2. Il tavolo di lavoro
Sul tavolo di lavoro non operate, ma potete osservare gli effetti delle operazioni compiute col joystick sul pannello di comando.

3. Pannello di comando
Il pannello di comando è costituito da tanti gadgets quadrati e rettangolari che vi consentono di modellare il paesaggio.
Azionando su di ognuno di essi sarete in grado di impartire le istruzioni necessarie a modificare la struttura del paesaggio (per essere più precisi dell'area visualizzata nel tavolo di lavoro), facendogli assumere la forma e la struttura che più vi piace.
Su ogni riquadro compare un disegno atto ad indicare l'effetto causato da un azione realizzata con quel gadget (ad esempio una freccia verso l'alto disegnata nel riquadro tenderà a far alzare il livello del terreno).
Così vi saranno gadgets che storceranno il terreno, altri che lo piegheranno a destra o a sinistra, altri che lo alzeranno o abbasseranno, altri ancora che creeranno avvallamenti e così via.
In pratica nessuna possibilità vi è preclusa e potrete modellare lo scenario come più vi piace.
Per poter azionare un gadget sarà sufficiente che lo selezionate con la leva del joystick (il gadget attivo è quello circondato da un riquadro) e che quindi premiate il tasto del fire.
Più tenete premuto il tasto e più l'effetto desiderato verrà accentuato.

4. Pannello di controllo
Tale pannello posto al limite inferiore dello schermo, propone una serie di comandi che, per lo più, ripete i controlli inseriti anche nei menu a tendina.
Ciò viene fatto in maniera da rendere più semplice e veloce l'accesso a certi comandi di uso frequente, come ad esempio il LOAD o il SAVE.
Tuttavia nel pannello di controllo appaiono anche altri comandi non compresi nei menu a tendina e che servono a definire gli attributi al paesaggio che si sta producendo o a consentirne una più facile realizzazione.
Così ad esempio certe aree di schermo possono essere memorizzate e ricopiate con un altro numero (per velocizzare il lavoro di creazione), oppure possono essere cancellate perchè non più necessarie.
I vari comandi possono essere raggiunti con la leva del joystick e premendo poi il tasto del fire.
Al momento del salvataggio (SAVE) dello scenario il computer vi chiederà un numero; a tale numero verrà associato il file di dati creato e dovrà esser ricordato quando lo stesso file dovrà essere ricaricato in memoria con il comando LOAD.

2) EDITOR 2

Anche in questo caso lo schermo è diviso in quattro parti:
1. La barra dei menu a tendina nella riga superiore dello schermo
2. Il tavolo di lavoro.
3. Il pannello dei comandi del tutto simile a quello dell'editor precedente.
4. Il pannello di controllo che anche qui ripropone i controlli che altrimenti sarebbero accessibili attraverso il menu a tendina.

1. La barra dei menu
La barra sulla riga superiore dello schermo consta di quattro menu a tendina: FILE, GENERAL, AREA e CONDITION.
Le modalità per accedere alla barra dei menu è la solita: basta spostare verso l'alto la leva del joystick; a questo punto vedrete evidenziarsi uno dei titoli dei menu a tendina: premendo il fire il menu calerà sullo schermo evidenziandone le voci; spostando invece lateralmente la leva del joystick si passerà ad evidenziare il titolo del menu a tendina seguente.
Con FILE vi si consente di operare sul disco, vale a dire di salvare o caricare il lavoro già fatto.
GENERAL permette di definire le condizioni

IL MENU DELL'EDITOR DELLE CONDIZIONI

IL MENU DELL'EDITOR DI AMBIENTE

generali del gioco, come ad esempio la maniera in cui si sposta la vostra nave sullo schermo.
Con AREA, invece, ci si può spostare da un area all'altra del campo di gioco.
CONDITION infine permette di definire le condizioni particolari di gioco, come ad esempio cosa succede se la navicella va a collidere contro un elemento del terreno.

2. Il tavolo di lavoro
Il tavolo di lavoro vi serve, come al solito, per osservare gli effetti delle operazioni compiute col joystick sul pannello di comando.

3. Pannello di comando
Il pannello di comando è del tutto simile a quello dell'editor precedente, infatti, anche i questa sede è possibile modellare il paesaggio, evitando di dover uscire dal programma e rientrare nella sezione precedente.

4. Pannello di controllo
Tale pannello posto al limite inferiore dello schermo, propone una serie di comandi che ancora ripete i controlli inseriti nei menu a ten-

dina.
Ciò viene fatto in maniera da rendere più semplice e veloce l'accesso a certi comandi di uso frequente, come ad esempio il LOAD o il SAVE.
Tuttavia nel pannello di controllo appaiono anche altri comandi non compresi nei menu a tendina e che servono a definire le condizioni di gioco, generali o particolari, oltre che a fornire utili strumenti di test del gioco che si sta realizzando.
Così ad esempio il comando TEST vi sarà utile per 'provare' il gioco ed apportare eventuali correzioni nel caso non siate soddisfatti del vostro lavoro.
Il comando RESET vi consente di annullare le disposizioni impartite per ridefinire tutto quanto dall'inizio.
Al momento del salvataggio (SAVE) dello scenario il computer vi chiederà un numero; a tale numero verrà associato il file di dati creato e dovrà esser ricordato quando lo stesso file dovrà essere ricaricato in memoria con il comando LOAD.

3) COMPILATORE

Una volta che avete 'dettato' le istruzioni per la realizzazione del gioco, non vi resta che ordinare al Commodore 64 di tradurre i vostri comandi nel codice che costituisce il videogame vero e proprio.
Infatti sino a questo punto avete detto al computer, in un linguaggio a voi congeniale, come deve essere fatto il gioco, questi ha registrato le vostre istruzioni, ma, per come sono ora, non è capace di leggerle e di tradurle in un programma funzionante.
Insomma va fatta la traduzione del gioco in linguaggio macchina.
Ecco allora che entra in scena il compilatore: questa è la fase di più facile esecuzione, infatti non dovrete far altro che seguire le istruzioni del computer, sarà poi lui a generare il codice macchina.
In effetti il compilatore si presenta con una sequenza di domande a cui dovrete rispondere, eventualmente compiendo qualche azione quale ad esempio l'inserimento di un dischetto dati nel drive.
Mi raccomando di non utilizzare come disco dati lo stesso in cui c'è il programma, ma di usarne sempre uno vuoto e già formattato, infatti il compilatore va a scrivere sul disco che avete nel drive e potrebbe rovinare importanti blocchi di memoria in cui è contenuto il codice del generatore di giochi.
Ad ogni buon conto il programma vi avverte e vi da il tempo per inserire nel drive un disco dati.
La sequenza di operazioni è completamente automatica, non mi sembra quindi il caso di doversi soffermare più di tanto, sappiate comunque che, al termine dell'operazione, il vostro gioco è registrato sul disco dati e ha il nome di 3D, quindi, per lanciarlo, dovrete impartire il comando:

LOAD '3D',8,1

Buon lavoro.

L'INTERFACCIA UTENTE

Nel programma che vi presentiamo avrete senz'altro notato la potenza dell'interfaccia utente. È chiaro che un più amichevole rapporto uomo-macchina non fa che aiutare lo sviluppo e l'utilizzo del computer e a testimonianza di ciò è l'attenzione posta dalle case di software nella produzione di programmi.
Eppure queste tecniche nuove hanno radici antiche, se è vero che gli studi effettuati a Silicon Valley risalgono ad un paio di decenni fa.
In questa sede vorrei ripercorrere la storia di questi studi, per capire quali siano state le direttive che hanno spinto gli studiosi a scegliere certe strade piuttosto che altre e a presentarci oggi dei prodotti che, rispetto a quelli di pochi anni orsono, risultano molto più semplici da usare e, per usare una parola un po' inflazionata, amichevoli.

Introduzione

Il colloquio uomo-computer è fortemente influenzato dalla predisposizione alla comunicazione dei due interlocutori e dalla maniera in cui la comunicazione deve avvenire.
Se è vero, infatti, che la macchina è sempre pronta a comunicare con l'uomo, è anche vero che non sempre l'uomo è in grado di comprendere i meccanismi di funzionamento della macchina e il linguaggio, strettamente vincolante, che si deve usare per fargli eseguire un lavoro.
Il calcolatore, che un tempo veniva usato solamente da esperti del settore, è entrato ora nelle case e negli uffici e viene adoperato da gente 'comune', non adusa al lavoro col computer e senza particolari conoscenze.
Il primo impatto con la macchina non può che essere traumatico: dover gestire un 'mostro' che ragiona in un modo tutto suo, che viene diretto attraverso un rigido linguaggio, che non è capace di ragionare con un pò di buon senso e, soprattutto, che si ferma se non gli si danno degli ordini precisi.
Non c'è da stupirsi, quindi, se il calcolatore è guardato con un pò di diffidenza e soggezione, non tanto da chi dirige le aziende in cui è stato inserito (dopo tutto il calcolatore aumenta notevolmente la produttività), ma soprattutto dai dipendenti ai livello i più bassi, quelle persone che lo devono usare.
Naturalmente dopo un certo lasso di tempo chiunque riesce ad impratichirsi con la macchina, a velocizzare il proprio lavoro e ad adeguarsi alla sua filosofia: il più delle volte questo avviene non perchè l'utente si rende conto delle proprie azioni, della necessità di impartire un certo comando o di compiere una certa operazione, ma perchè ha imparato che 'così si deve fare' e prosegue il suo lavoro senza saper precisamente come la macchina opera.
Del resto, chiunque abbia avuto la possibilità di usare il calcolatore nel suo lavoro, si è reso conto di quanto sia accresciuto il proprio rendimento e di come le operazioni diventino più semplici e sbrigative; certamente nessuno auspicherebbe un ritorno ai metodi tradizionali, che sarebbero considerati troppo lenti ed onerosi.
I programmi applicativi, specie quelli destinati ad un pubblico poco esperto nell'uso del computer, sono solitamente realizzati tenendo conto di queste considerazioni ed instaurano un colloquio uomo-macchina il più possibile guidato, cercando di limitare al massimo l'iniziativa dell'utente e facendo operare sotto uno stretto controllo, che non consenta il verificarsi di errori irrimediabili.
L'input dei dati è sempre controllato e sostenuto da messaggi di aiuto, le scelte delle operazioni possibili avvengono sempre attraverso dei menu, evitando così la necessità di dover digitare direttamente il comando e limitando in questo modo la probabilità di errore.
Resta comunque il rammarico di non poter usare il computer in una maniera più semplice e naturale, senza cioè dover sottostare alle strette regole di conversazione e ai meccanismi di funzionamento della macchina.
Proprio per migliorare questo aspetto del rapporto uomo-macchina, le aziende che producono software hanno cercato di creare i presupposti per attuare una rivoluzione in questo campo; era infatti necessario studiare a fondo il problema al fine di identificare quali strumenti (hardware e software) potessero in qualche modo aiutare l'utente finale a trovare un ambiente operativo meno ostico.
Il risultato di queste ricerche è costituito da un insieme di strumenti che si sono dimostrati

molto utili nel rendere un computer più facilmente usabile, non solo dall'utente in esperto, ma anche da coloro che lavorano nel campo dell'informatica e non avevano in realtà sentito alcun bisogno di innovamento.

I primi studi per un nuovo tipo di interfaccia utente.

Gli studi che hanno portato alla definizione di nuove e più moderne tecniche di interfacciamento tra calcolatore ed utente sono stati influenzati molto dall'ideazione e dallo sviluppo di nuove componenti hardware, che hanno in qualche modo costituito la base tecnologica su cui realizzare un software più semplice da usare.
Era infatti necessario creare dei nuovi presupposti hardware perchè si potessero apportare dei miglioramenti significativi al software, che, per la verità, è sempre stato realizzato con l'intento di venire incontro alle esigenze dell'utente finale.
Lo sforzo dei ricercatori è stato quello di concentrarsi sulla maniera in cui la gente lavora piuttosto che su quella in cui lavorano i computer.
Le persone sono abituate a maneggiare oggetti, a prenderli, a spostarli, a buttarli via quando non servono più; inoltre spesso nel lavoro, per compiere una certa mansione, è necessario consultare dei dati, senza però perdere di vista le informazioni che in quel momento sono sul video; infine, per un utente, è molto più semplice indicare una scelta di un menu esplicitamente (se è possibile con le dita), piuttosto che usare la tastiera come intermediario e digitare una lettera o un numero.
Se si vogliono trattare degli elementi posti sullo schermo, per prenderli, spostarli, appoggiarli da qualche parte e gettarli nel cestino quando non li si vuole più usare, è necessario disporre di un video ad alta risoluzione, possibilmente a colori e su cui si possa esercitare un controllo pixel per pixel; ci deve essere un dispositivo di puntamento capace di indirizzare ogni singolo punto grafico e si deve avere la facoltà di dividere lo schermo in tante porzioni su cui poter caricare applicazioni e dati diversi, per poterli consultare intanto che si lavora.
Di tutto questo se ne era già accorto Douglas Englebart, che aveva a lungo studiato il comportamento umano di fronte ad un calcolatore, prima di inventare il mouse, il dispositivo di puntamento che opera pixel per pixel ed è quindi indispensabile per individuare aree ben delimitate dello schermo.
Ma anche altri studiosi si sono occupati di questo problema ed hanno condotto indagini statistiche atte ad individuare l'efficenza degli utenti quando questi usano certi strumenti di lavoro, l'aumento della loro produttività, la diminuzione degli errori nell'impartire i comandi ecc.
La prima grande casa che ha però dedicato energie e risorse allo studio di queste problematiche è stata la Xerox PARC (Palo Alto Research Center) che nel 1971 ha unito un gruppo di ricercatori sotto il nome di LRG (Learning Research Group), con a capo Alan Kay.
Questo gruppo di ricercatori è stato il primo a definire gli strumenti software che assicurano 'l'illusione utente', locuzione usata per indicare i programmi che fanno da intermediari tra le applicazioni e il calcolatore, rendendo più snello e piacevole il lavoro.
Lo scopo dell'illusione è insomma quella di aumentare le capacità di simulazione dell'utente, dandogli l'impressione di lavorare con oggetti flessibili che soddisfino le proprie esigenze.
Dagli sforzi di questi ricercatori nacquero quindi nuovi concetti su come l'utente debba interagire col calcolatore, tali teorie vennero poi messe in pratica con la definizione del linguaggio di programmazione Smalltalk il primo ambiente operativo in cui si facesse uso di finestre, e con la creazione della workstation XeroxStar con la quale comparvero per la prima volta le icone.

La nuova tecnologia

Come è già stato sottolineato in precedenza, in questo campo le innovazioni hanno fatto seguito e sono state subordinate all'invenzione di nuovi dispositivi di input/output, quali, ad esempio, il mouse, il joystick, la penna, le tavolette grafiche e altri, che, più che sostituire la tastiera, ineliminabile per taluni scopi, gli si sono affiancati, arricchendo il sistema e facilitando alcune operazioni.
La caratteristica comune a tutte queste device è che sono tutte in grado di indirizzare i punti grafici dello schermo e sono perciò chiamate pointing devices.
La tavoletta grafica è uno dei dispositivi di puntamento a più basso costo; consiste appunto in una tavoletta collegata al calcolatore sulla quale si può far agire con un dito una bacchetta appuntita ed avere un corrispondente movi-

mento di una freccetta sullo schermo.
In pratica la tavoletta converte costantemente le sue proprie coordinate in coordinate dello schermo e, per mezzo di un tasto, permette di selezionare ogni singolo pixel.
Una delle tavolette grafiche che ha riscosso un buon successo è la Koalapad, che viene tuttavia usata più per la produzione di disegni tecnici che come semplice dispositivo di puntamento.
Il touchscreen corrisponde ad una tavoletta grafica in cui è lo schermo stesso a costituire il piano su cui muovere le dita.
Naturalmente è un dispositivo un pò più costoso delle tavolette grafiche, ma ha il non trascurabile pregio di permettere un puntamento diretto e non analogico degli elementi sul video; infatti il monitor del sistema è coperto di una pellicola trasparente e sensibile che rileva la posizione in cui viene appoggiato il dito, e la converte nelle coordinate del punto sottostante.
Simile al touchscreen è il sistema di puntamento che fa uso di una penna elettronica (lightpen), con la differenza che l'utente, in questo caso, deve utilizzare al posto delle dita una piccola bacchetta elettronica.
Questo metodo, come quello precedente, ha il vantaggio di non richiedere dello spazio supplementare sul tavolo di lavoro per appoggiare gli strumenti di puntamento.
Più simili alle tavolette grafiche sono le tavolette digitali (digitizing tablets) che sono in tutto simili alle precedenti, ma, necessariamente, fanno uso di una piccola penna elettrica.
Ciò comporta alcuni inconvenienti per un certo tipo di lavoro, ma assicura una accurata fedeltà nel tradurre le locazioni della tavoletta in coordinate dello schermo.
Le trackballs sono palle all'interno di una scatoletta con il lato superiore aperto per far affiorare la sfera e per permettere all'utente di farla rotolare in ogni senso.
Collegata al movimento della palla è la frecciolina sullo schermo, che può essere così usata per indicare i punti grafici del video.
I joystick sono invece dei dispositivi a bassissimo costo, composti da una scatola da cui esce un bastoncino che può essere spostato in varie posizioni e permette il movimento di un cursore sullo schermo.
Queste ultime due soluzioni sono state provate come dispositivi di puntamento per un ambiente di lavoro, ma si sono rivelate piuttosto carenti: prima di tutto non consentono un indirizzamento del cursore con una sufficiente precisione, inoltre, per quanto riguarda il joystick, gli spostamenti avvenivano con troppa lentezza e questa è una cosa che infastidisce notevolmente l'utente.
Dal canto loro le trackballs sono abbastanza veloci come dispositivi di puntamento, ma, meccanicamente, il lavoro diventa troppo dispendioso ed affaticante per l'utente, se si pensa che questi è costretto, per diverse ore al giorno, ad armeggiare con una palla; senza contare che, inevitabilmente, l'utente sfrega le dita contro i bordi della scatola nei punti in cui la sfera vi rientra e ciò, a lungo andare, causa degli arrossamenti e dolori alle mani.
Questi strumenti sono invece risultati molto validi come dispositivi di comando nei videogames ed in questo campo vengono infatti largamente usati.
Il mouse è una piccola scatola collegata al computer per mezzo di un cavetto; sulla parte inferiore, a contatto col piano di appoggio, c'è una sferetta che, se fatta rotolare, aziona un cursore sullo schermo e permette il puntamento dei pixel; sulla parte superiore invece vi sono due o più pulsanti che servono ad impartire comandi o a selezionare le aree raggiunte dal cursore.
Il mouse è sicuramente il dispositivo di puntamento che ha riscosso il maggior successo e che viene più spesso usato.
In effetti questo strumento offre notevoli vantaggi rispetto a quelli presentati fino ad ora e soprattutto non è particolarmente indicato per un certo tipo di lavoro, ma è una device piuttosto duttile che ben si adatta a tutti gli ambienti operativi.
Ad esempio le tavolette digitali sono particolarmente indicate per il disegno tecnico e per un loro uso nell'area delle applicazioni CAD e CAM, ma non vanno per niente bene per una segretaria che deve continuamente usare la tastiera per battere un testo od introdurre i dati, perchè deve sempre appoggiare e riprendere la penna con cui disegnare sulla tavoletta.
Lo stesso si può dire per le penne elettroniche che vanno incontro al medesimo inconveniente. Inoltre sia le penne elettroniche che i touchscreen, a lungo andare, provocano un affaticamento per l'utente, costretto ad alzare continuamente il braccio per puntare degli elementi sullo schermo, senza contare che durante questa operazione le mani e le braccia possono nascondere parti del video e quindi rendere più problematico il lavoro.

Un altro difetto di penne elettroniche e touchscreen è quello di richiedere la presenza dell'utente molto vicino al monitor in modo che questi lo possa toccare per puntare gli oggetti; ciò causa un affaticamento degli occhi ed espone in maggior misura l'utente alle radiazioni emesse dal video.
Esistono tuttavia campi di applicazione in cui queste due strumenti rappresentano la pointing device ideale, come ad esempio i sistemi di information retrieval posti in luoghi pubblici dove l'utente è una persona di passaggio che compie un numero limitato di interrogazioni.
Una situazione simile si può verificare ad esempio nelle stazioni ferroviarie equipaggiate con terminali pubblici per la consultazione degli orari di partenza dei treni, oppure nei saloni fieristici in cui un visitatore può richiedere la locazione di un particolare stand o le caratteristiche dei prodotti esposti.
Il mouse elimina gli inconvenienti sopraindicati, assicura una notevole precisione ed una discreta velocità nel puntamento, mantiene fissata la posizione del cursore qualora si stacchi la mano per usare la tastiera ed è anche piuttosto indicato per compiere disegni sullo schermo.
Col mouse l'input avviene sempre attraverso il cursore ad esso associato (in genere una piccola freccia), che seleziona una parte dello schermo, ad esempio una voce di un menu, colorandola di un altro colore e preparandola per essere scelta, premendo il tasto del mouse l'entità selezionata viene inviata come dato di input da essere processato dal sistema.
A differenza di tutti gli altri richiede oltre ad un certo spazio per poterlo muovere sulla scrivania, anche una periodica pulizia perchè 'ingerisce', attraverso la pallina, la polvere che si trova sul piano di appoggio.
Tutti questi dispositivi, mouse compreso, risultano più lenti nell'impartire i comandi se comparati con sequenze di controllo digitate attraverso la tastiera da un buon dattilografo; è infatti più veloce premere due tasti in successione (ESCE) piuttosto che prendere il mouse, spostare il cursore sopra il comando voluto e premere il tasto, ma ciò va contro i presupposti che sono stati indicati all'inizio, cioè che per l'utente è molto più semplice e naturale impartire un comando vedendo quello che si fa, piuttosto che digitare oscure sequenze di caratteri che, per di più, lo espongono al costante rischio di errori.
Inoltre non tutti gli utenti sono buoni dattilografi o hanno voglia di memorizzare tutte le sequenze di controllo necessarie per far girare un programma, anche perchè, per applicazioni piuttosto complesse, i comandi possibili possono anche essere molto numerosi.
Questi dispositivi di puntamento possono essere utilizzati solamente con schermi ad alta risoluzione a colori di tipo Raster, su cui possono essere eseguiti disegni e figure sofisticate, che rendono il primo impatto tra programma e utente meno traumatico rispetto agli schermi in bassa risoluzione o a quelli predisposti al semplice display di testi.
Certo per l'utente una cosa è veder comparire il cursore lampeggiante in attesa di una linea di comando, un'altra è invece una figura colorata, con una freccia comandata dal mouse che può raggiungere qualsiasi pixel dello schermo e che quando si sovrappone ad un comando (già evidenziato sul video e che non necessita di essere digitato) lo dipinge di un altro colore per evidenziare la selezione.
Inoltre, con monitor di questo tipo, si cerca sempre di visualizzare le informazioni nella maniera più gradevole possibile, rispondendo istantaneamente ai comandi dell'utente: se egli intende spostare un oggetto (un'icona o una finestra) che compare sul video, il sistema deve reagire immediatamente, dando l'impressione a chi lo sta usando, di essere sempre padrone della situazione.
Questi risultati possono essere raggiunti solo con l'ausilio di processori veloci a 16 o a 32 bit: è chiaro infatti che il software che governa tali sistemi è sensibilmente più complesso ed il processore deve essere in grado di trattare con scioltezza le grosse bitmap che mantengono in memoria la copia dell'immagine video.
Inoltre questi sistemi di elaborazione necessitano di una quantità abbastanza elevata di memoria centrale, ma questo non è più un grosso problema, visto che la tendenza è quella di garantire sempre più a buon mercato le espansioni di memoria.
Da tutto ciò ci si rende conto del fatto che i risultati raggiunti nell'ideazione e nella realizzazione di interfacce utente più sofisticate non sono semplicemente stati il frutto di studi compiuti sul comportamento umano di fronte ad un calcolatore, ma sono stati resi possibili dal verificarsi di un contemporaneo sviluppo tecnologico nel campo della grafica ad alta risoluzione, dei microprocessori e dei nuovi dispositivi di puntamento capaci di indirizzare i singoli pixel.

Infine si può anche citare la metamorfosi che ha subito la tastiera, che da semplice replica di una macchina per scrivère si è via via arricchita di nuovi tasti e ha cambiato la posizione di quelli vecchi, cercando di rendere più confortevole la digitazione dei dati.
Una disposizione ergonomicamente studiata dei tasti aiuta l'utente a commettere meno errori di battitura; i produttori di computer hanno cercato di analizzare la frequenza con cui gli utenti usavano i vari tasti ed hanno deciso quali tra essi potevano essere raggiunti solo attraverso la pressione contemporanea dello shift o del control.
Sono comparsi i tasti funzionali, il cui scopo è di carattere esclusivamente ergonomico, perchè sono associati ai comandi che vengono usati più frequentemente e ne permettono l'esecuzione con una semplice pressione.
In genere, insieme ad essi, i nuovi personal computer rendono disponibile un tastierino numerico, che consente l'inserimento di dati aritmetici con una maggior velocità e i tasti di indirizzamento del cursore, che permettono all'utente di muoverlo in maniera discreta a qualsiasi locazione del video.
Lo scopo di tutto ciò è come al solito quello di rendere più confortevole l'uso di questo dispositivo di Input/Output che, per certi versi, resta una periferica di importanza primaria per la comunicazione tra l'utente e la macchina.
Non ci si deve dimenticare che ogni comando impartito col mouse o con qualsiasi altra device può essere replicato con una sequenza di controllo della tastiera, che tale operazione è in genere più veloce (anche se meno immediata e meno naturale), e che sicuramente, dopo che l'utente si è un pò impratichito col programma, la tastiera comincia a sostituire le altre device di Input/Output per la segnalazione di certi comandi.

Gli strumenti software

Gli strumenti software che realizzano questa nuova generazione di interfacce utente sono abbastanza complesse e fanno un largo uso dei colori e della grafica ad alta risoluzione.
Come detto l'utente vorrebbe poter aver a portata di mano più dati ed applicazioni possibili, per potersi consultare od ottenere risultati di un programma mentre sta lavorando con un altro.
Far girare due programmi contemporaneamente non è possibile se si lavora con un calcolatore solo (a meno che questi non funzioni in multitasking), ma non è neanche possibile osservare due applicazioni diverse su uno stesso monitor, perchè ognuna di esse richiede l'uso esclusivo del dispositivo di output.
Per questa ragione se certi dati sono prodotti da un programma, e un utente li vuole consultare durante l'esecuzione di un altro programma, l'unica via da seguire è quella di provvedere a stampare su carta i dati desiderati, prima di incominciare a lavorare con la seconda applicazione.
Una delle novità delle interfacce utente moderne è quella di dividere lo schermo in più sezioni rettangolari chiamate finestre, al cui interno viene fatta girare un'applicazione diversa ed indipendente da quelle delle altre finestre.
Il video diventa così uno spazio utile a contenere tanti monitor virtuali che l'utente può aprire e chiudere a piacere.
Come detto in ognuna delle finestre può essere fatta girare un'applicazione, ma solo se il sistema operativo funziona in multitasking i vari programmi lavorano contemporaneamente, altrimenti sarà quello che è contenuto in una finestra particolare, chiamata finestra attiva, che potrà disporre della CPU per portare avanti l'elaborazione.
Naturalmente l'utente ha la facoltà di rendere attiva l'una o l'altra finestra presente sul video, eseguendo così un continuo cambiamento di contesto.
Le finestre possono poi essere spostate, ingrandite o rimpicciolite e si possono sovrapporre l'una all'altra proprio come dei fogli di carta appoggiati sulla scrivania; può anche accadere che una finestra nasconda completamente altre finestre sotto di essa e sia perciò necessario diminuirne le dimensioni.
Ci si rende conto di quanto questi strumenti aiutino chi lavora col calcolatore: si pensi ad una persona che sta usando un word processor e possa caricare sullo schermo (all'interno di diverse finestre) tanti documenti, per potersi consultare durante la stesura di uno scritto; o ancora a chi debba lavorare con un foglio elettronico ed abbia bisogno di dati contenuti in un database.
Lo spazio riservato ad una finestra è generalmente inferiore a quello dell'intero video, ma per ognuna di esse c'è la possibilità di effettuare uno scroll orizzontale e verticale, aiutandosi col mouse ed usando degli speciali strumenti, chiamati scrollbar, che stanno a fianco

della stessa.
Ogni finestra è sovrastata da una barra orizzontale su cui viene scritto il nome dell'applicazione contenuta all'interno.
Un altro strumento che contribuisce a migliorare notevolmente l'interazione tra l'utente e il calcolatore è il pull down menu (chiamato anche popup menu).
Si presenta sullo schermo come una fascia orizzontale su cui è indicato l'elenco dei menu disponibili per l'applicazione in corso; per accedervi è sufficiente posizionarsi col cursore del mouse sopra al nome del menu interessato, si vedrà allora che dalla barra del menu scenderà una lista di opzioni selezionabili, ognuna di esse svolge un'operazione particolare che può essere scelta se ci si pone sopra di essa col cursore e si preme il tasto del mouse.
I vantaggi di questo sistema sono evidenti:
- sul video i menu occupano normalmente solo una riga, invece di utilizzare l'intera pagina o di non comparire affatto.
- tutti i menu sono raggiungibili direttamente col mouse ed in maniera veloce, senza dover superare un certo numero di passi e di schermate per raggiungere il menu e la voce desiderata.
- la lista dei menu contiene solo le opzioni utili in quella fase di lavoro, senza confondere l'utente con lunghe ed in gran parte obsolete liste di comandi.
- le voci del menu non oscurano completamente la finestra che contiene l'applicazione, è così possibile osservare contemporaneamente il menu e il lavoro.
- per eseguire una scelta non si deve digitare un tasto (un numero o una lettera) che rappresenta una voce di un menu, ma si seleziona direttamente l'opzione desiderata indicandola col mouse e riducendo così le probabilità di errore.
- si può eseguire una scelta senza dover spostare lo sguardo dal video alla tastiera, perchè il mouse è uno strumento di Input/Output che non necessita di essere osservato durante l'uso.

Le dialog box sono strumenti rettangolari di cui il sistema si serve per comunicare informazioni rapide all'utente.
Esistono due tipi di dialog box:
- quelle appartenenti alla prima specie sono molto simili alle finestre ma non racchiudono alcuna applicazione, solo dei messaggi per avvertire l'utente che sta compiendo degli errori o che il comando digitato determina il cancellamento di un file; per questa ragione vengono chiamate anche message box.

L'apparizione di un message box ferma l'applicazione in corso, che non viene ripresa fino a che l'utente non ha letto e risposto al sistema; questo garantisce che egli abbia capito il messaggio e sappia sempre che cosa sta facendo il computer, specialmente per delle operazioni critiche come la cancellazione di un file.
- le dialog box della seconda specie, invece, servono a ricevere dell'input veloce ed inderogabile dall'utente; ad esempio il nome con cui si vuole salvare un certo file, o le modalità con cui si vuol stampare un documento.

Perciò le dialog box possono avere dei campi da riempire o dei 'tasti' software che compaiono premuti o rilasciati per indicare l'attivazione o meno di una certa opzione.
In generale il pregio delle dialog boxe è quello di attirare l'attenzione molto di più dei messaggi di errore tradizionali, perchè compaiono in mezzo allo schermo e possono essere corredate da disegni (punti interrogativi, segnali di stop, ecc.) che fanno ben capire che qualcosa che non va nell'operazione che si sta tentando di compiere.
Le icone sono un'altra facility delle nuove interfacce utente.
Una figura viene sicuramente recepita più velocemente rispetto ad una scritta: partendo da questo presupposto e sfruttando le potenzialità della grafica ad alta risoluzione, lo schermo può essere riempito di piccoli disegni, che rappresentano ad esempio le periferiche del sistema, e che possono essere raggiunte, selezionate e richiamate col mouse.
Un programma gestionale può avere un'icona della stampante che, se selezionata, fa partire il programma di stampa di un file.
Oppure ci può essere l'icona del drive, che può essere 'aperto' per avere sullo schermo la lista dei files in esso contenuti.
Ma l'uso delle icone non è limitato alla rappresentazione delle periferiche: ad esempio ogni programma contenuto in un dischetto può essere raffigurato sullo schermo con un'icona particolare, che richiami alla memoria la natura dell'applicazione e che sia così più facilmente individuabile in mezzo agli altri programmi.
Addirittura c'è la possibilità di associarle ad un comando: l'icona di un bidone della spazzatura può essere piazzata in un punto dello scher-

mo e può essere associata al comando di delete; quando l'utente vuole cancellare un file, lo deve raggiungere col mouse, lo deve selezionare e trasportare nel bidone, il sistema gli chiederà con una dialog box se è sicuro di voler cancellare quel file e, se l'utente risponde affermativamente, processerà il comando.
Con questa procedura è più difficile che una persona distratta compia degli errori così drastici come il cancellamento per sbaglio di un programma.
Da ultimo le icone possono essere usate a fini esclusivamente estetici, per evidenziare sul video il logotipo del programma o della software house che l'ha prodotto, oppure per riempire il video con figure che rendano più piacevoli le varie schermate.
Sullo schermo possono poi essere inseriti vari tasti o bottoni software, con lo scopo di rendere più facilmente selezionabili le opzioni del programma; l'idea è ancora quella che sta dietro al concetto di icona: un oggetto visibile e raggiungibile col mouse, per un utente, è sicuramente più gradevole di una combinazione di tasti da digitare per ottenere lo stesso scopo.
Inoltre i bottoni software sono autoesplicativi, accanto o sopra essi vi è una scritta che indica l'opzione associata e rimangono selezionati fintanto che l'utente non li disattiva; ve ne è poi un particolare tipo, chiamato radio button, i cui elementi si disattivano a vicenda nel caso che le opzioni che rappresentano siano mutuamente esclusive.

Ciò che rende queste interfacce veramente potenti è la possibilità di far condividere i dati di due finestre (applicazioni) diverse.

Questa caratteristica potenzia notevolmente il sistema e fa dell'utente il vero padrone dei dati, si pensi a due finestre che contengono due documenti e si voglia trasportare una porzione di un documento all'interno dell'altra finestra: il sistema permette di selezionare col mouse un blocco di testo, memorizzarlo in un buffer, cambiare contesto attivando la seconda finestra ed inserire il blocco di testo memorizzato in un qualsiasi punto del secondo documento.

Questo può essere fatto nell'ambito dello stesso programma di Word Processing, ma è possibile lo scambio di dati anche tra applicazioni diverse ed indipendenti, come un Data Base ed un foglio elettronico.

Questa operazione di scambio dei dati è denominata Cut & Paste, perchè consente di 'ritagliare' una parte dell'area dati di un programma per 'incollarla' in un'altra applicazione.

Il buffer destinato a ricevere tali dati è a volte associato ad una icona che rappresenta una cartella (clipboard), con la solita intenzione di rendere palese all'utente la memorizzazione dei dati: ogni volta che si vogliono inserire dei dati nel buffer, l'utente dovrà selezionarli e trascinarli col mouse all'interno della clipboard, dove resteranno fino a che non si deciderà di eliminarli o di inserirli in un'altra applicazione.